

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No. : 10/015,958 Confirmation No. 6989
Appellant : David George De Vorchik *et al.*
Filed : October 30, 2001
Group Art Unit: 2192
Examiner : Eric B. Kiss
Title : METHOD AND SYSTEM FOR CHAINING AND EXTENDING
WIZARDS
Docket No. : 164122.01/MFCP.88142
Customer No. : 45809

EFS – 03 August 2007

APPELLANTS' APPEAL BRIEF

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This is an Appeal from a Final Office Action mailed 05 December 2006, rejecting claims 1-20. These claims have been at least twice rejected. Appellants, having filed a Notice of Appeal within the time period provided under 37 C.F.R. § 41.31(a) accompanied by the fee set forth in 37 C.F.R. § 41.20(b)(1), do hereby submit this Appeal Brief along with the fee set forth in 37 C.F.R. § 41.20(b)(2). The Commissioner is hereby authorized to charge any additional fee that may be due, or credit any overpayment, to Deposit Account No. 19-2112.

Contents

I.	Real Party in Interest.....	3
II.	Related Appeals and Interferences.....	4
III.	Status of Claims.....	5
IV.	Status of Amendments.....	6
V.	Summary of Claimed Subject Matter	7
	Claim 1	7
	Claim 7.....	7
	Claim 8.....	8
	Claim 9.....	8
	Claim 10.....	9
	Claim 11.....	9
	Claim 14.....	10
VI.	Grounds of Rejection to be Reviewed on Appeal.....	11
VII.	Argument	12
	A) Rejection under 35 U.S.C. § 102(b) over Fedorov et al. (Professional Active Server Pages 2.0,) 1998, Wrox Press Ltd)	12
	(i) Claims 7, 10, and 11	12
	B) Rejection under 35 U.S.C. § 102(e) over Gauthier <i>et al</i> (U.S. Patent No. 6,574,791).	16
	(i) Claims 1, 8, and 9.....	16
	(ii) Claim 2	17
	(iii) Claim 3.....	19
	(iv) Claim 14.....	20
	C) Rejection under 35 U.S.C. § 103(a) over Fedorov (Professional Active Server Pages 2.0,” 1998, Wrox Press Ltd) and Gauthier <i>et al</i> (U.S. Patent No. 6,574,791).	21
	(i) Claims 16 and 18-19	22
VIII.	Claims Appendix	25
IX.	Evidence Appendix	31
X.	Related-Proceedings Appendix.....	32

I. REAL PARTY IN INTEREST

The real party in interest is Microsoft Corporation, a corporation of the State of Washington, United States of America.

II. RELATED APPEALS AND INTERFERENCES

None.

III. STATUS OF CLAIMS

Claims 1-3, 7-11, 14, 16, and 18-19 are pending, and the rejection of each of those claims is being appealed.

IV. STATUS OF AMENDMENTS

An amendment was filed on 05 July 2007, subsequent to the Final Office Action dated 05 January 2007. The amendment canceled claims 4-6, 12, 13, 15, 17, and 20. An Advisory Action mailed 02 August 2007 confirmed that claims 4-6, 12, 13, 15, 17, and 20 are canceled. A listing of all claims currently pending is reproduced in the Claims Appendix.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The instant Application includes seven independent claims: claims 1, 7-11, and 14.

Claim 1

Claim 1 defines a method for use in a computing environment for extending a wizard. In accordance with the method, a host-wizard component (220) is provided. *Appellants' specification, at p. 11, ll. 15-20.* One or more sub-wizard components (230, 240) are also provided. *Appellants' specification, at p. 12, ll. 6-12.* During execution of the host-wizard component (220), the one or more sub-wizard components (230, 240) are invoked. *Appellants' specification, at p. 12, ll. 17-21.* In turn, control is transferred from said host-wizard (220) to said one or more sub-wizard components (230, 240). *Appellants' specification, at p. 13, ll. 3-5.*

Claim 7

Claim 7 defines a method for use in a computing environment for extending a wizard. In accordance with the method, a host-wizard (220) that defines an extension interface to respond to navigation events is generated. *Appellants' specification, at p. 14, ll. 14-20.* Additionally, a web component is generated that comprises a web page (400) containing a header area (402), a wizard control area (404), one or more object module functions (410, 412) to enable navigation, and a control interface area (406) having navigation control adapted to recursively navigate within said web component and to said host wizard by utilizing said one or more object module functions (410, 412). *Appellants' specification, at p. 15, l. 21-p. 17, l. 5.* In turn, a user interface that integrates said web component into said host wizard by utilizing the extension interface to perform recursive navigation between said web component and said host wizard is generated. *Appellants' specification, at p. 16, ll. 14-16.* The web

component and host wizard (220) exchange informational items via an information container.
Appellants' specification, at p. 17, ll. 6-13.

Claim 8

Claim 8 defines a machine readable medium having machine executable instructions for performing a method for extending a wizard. In accordance with the machine executable instructions, a host-wizard component (220) is generated. *Appellants' specification, at p. 11, ll. 15-20.* One or more sub-wizard components (230, 240) are also generated. *Appellants' specification, at p. 12, ll. 6-12.* During execution of the host-wizard component (220), the one or more sub-wizard components (230, 240) are invoked. *Appellants' specification, at p. 12, ll. 17-21.* In turn, control is transferred from said host-wizard (220) to said one or more sub-wizard components (230, 240). *Appellants' specification, at p. 13, ll. 3-5.*

Claim 9

Claim 9 defines a computer system having processor, a memory, and an operating environment, the computer system executes a method for use in a computing environment for extending a wizard. In accordance with the method, a host-wizard component (220) is generated. *Appellants' specification, at p. 11, ll. 15-20.* One or more sub-wizard components (230, 240) are also generated. *Appellants' specification, at p. 12, ll. 6-12.* During execution of the host-wizard component (220) the one or more sub-wizard components (230, 240) are invoked. *Appellants' specification, at p. 12, ll. 17-21.* In turn, control is transferred from said host-wizard (220) to said one or more sub-wizard components (230, 240). *Appellants' specification, at p. 13, ll. 3-5.*

Claim 10

Claim 10 defines a machine readable medium having machine executable instructions for performing a method for extending a wizard. In accordance with the machine readable instructions, a host-wizard (220) that defines an extension interface to respond to navigation events is generated. *Appellants' specification, at p. 14, ll. 14-20.* Additionally, a web component is generated that comprises a web page (400) containing a header area (402), a wizard control area (404), one or more object module functions (410, 412) to enable navigation, and a control interface area (406) having navigation control adapted to recursively navigate within said web component and to said host wizard by utilizing said one or more object module functions (410, 412). *Appellants' specification, at p. 15, l. 21-p. 17, l. 5.* In turn, a user interface that integrates said web component into said host wizard by utilizing the extension interface to perform recursive navigation between said web component and said host wizard is generated. *Appellants' specification, at p. 16, ll. 14-16.* During the recursive navigation, the web component and host wizard (220) exchange informational items via an information container. *Appellants' specification, at p. 17, ll. 6-13.*

Claim 11

Claim 11 defines a computer system having a processor, a memory, and an operating environment. The computer system includes a host-wizard (220) that defines an extension interface to respond to navigation events. *Appellants' specification, at p. 14, ll. 14-20.* Additionally, the computer system includes a web component which comprises a web page (400) containing a header area (402), a wizard control area (404), one or more object module functions (410, 412) to enable navigation, and a control interface area (406) having navigation control adapted to recursively navigate within said web component and to said host wizard by

utilizing said one or more object module functions (**410, 412**). *Appellants' specification*, at p. 15, l. 21-p. 17, l. 5. The computer system includes a user interface that integrates said web component into said host wizard by utilizing the extension interface to perform recursive navigation between said web component and said host wizard. *Appellants' specification*, at p. 16, ll. 14-16. The computer system includes, during the recursive navigation, an information container to exchange information items between the web component and host wizard (**220**). *Appellants' specification*, at p. 17, ll. 6-13.

Claim 14

Claim 14 defines a computer system having a processor, a memory and an operating environment, the computer system operable to execute a method for use in a computing environment for chaining wizards. *Appellants' specification* at p. 2, ll. 8-10. The computer system includes a first wizard (**220**) having panels (**204, 206**) to guide a user through a first task and a second wizard (**230, 240**) having panels (**208, 214**) to guide a user through a second task. *Appellants' specification* at p. 11, l. 20-p. 12, l. 20. Also, the computer system includes at least one navigation component (**200a**) on each of said first and second wizards, the navigation components allowing sequential progression or regression through said first and second wizards to chain said second wizard to said first wizard to guide a user through the first and second tasks. *Appellants' specification* at p. 12, ll. 18-23.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A) Whether claims 7, 10, and 11 are anticipated under 35 U.S.C. § 102(b) by Fedorov *et al.* (Professional Active Server Pages 2.0," 1998, Wrox Press Ltd).

B) Whether claims 1-3, 8, 9, and 14 are anticipated under 35 U.S.C. § 102(e) by Gauthier *et al* (U.S. Patent No. 6,574,791).

C) Whether claims 16 and 18-19 are anticipated under 35 U.S.C. § 103(a) by Fedorov *et al.* (Professional Active Server Pages 2.0," 1998, Wrox Press Ltd) and Gauthier *et al* (U.S. Patent No. 6,574,791).

VII. ARGUMENT

A) Rejection under 35 U.S.C. § 102(b) over Fedorov et al. (Professional Active Server Pages 2.0.) 1998, Wrox Press Ltd)

“A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.” *Verdegall Brothers v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). “The identical invention must be shown in as complete detail as is contained in the . . . claim.” *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). *See also*, MPEP § 2131.

Claims 7, 10, and 11 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Fedorov. Appellants respectfully traverse this rejection, as hereinafter set forth.

(i) Claims 7, 10, and 11

Appellants respectfully submit, with respect to independent claims 7, 10, and 11, Fedorov fails to disclose, among other things, “generating a host wizard that defines an extension interface to respond to navigation events; and generating a user interface that integrates said web component into said host wizard by utilizing the extension interface to perform recursive navigation between said web component and said host wizard.”

Appellants submit that the Office’s interpretation of “wizard” as utilized in claims 7, 10, and 11 and defined in the specification, at p. 2, ll. 7-10, goes beyond the reasonable ordinary and customary meaning of the term as understood by an artisan of ordinary skill. *Philips v. AWH Corp.*, 415 F.3d 1303, 1313, 75 USPQ2d 1321, 1326 (Fed. Cir. 2005). In *Philips*, the court noted “[i]mportantly, the person of ordinary skill in the art is deemed to read

the claim term not only in the context of the particular claim in which the disputed term appears but in the context of the entire patent, including the specification. *Id.* at 1313.

In the specification, at p. 2, ll. 7-10, Appellants define “wizard” to mean an application that directs a user through a configuration process or the implementation of a particular task within an application program; a “wizard” is essentially a programmatic method of providing guidance to a user within a controlled environment and in a predictable manner. The programmatic method includes a multistep process that is controlled by a user’s navigation of screens to answer questions and ultimately complete an operation.

The Office cites *Schulze* to suggest that counsel’s arguments with respect to interpretation of the “wizard” should not take the place of evidence in the record. *In re Schulze*, 346 F.2d, 600, 145 USPQ 716. In *Schulze*, the Court was presented with a patent application for a method for coating fine aggregates, such as finely-divided coat particles, with a thin bituminous film. On appeal, the appellant argued the importance of specific temperatures and pressures for solids, water, and bitumen. The court reasoned that appellants’ assertion that limitations regarding the importance of specific temperatures and pressures were not supported by the specification because the record, i.e. the specification, failed to provide the patentable significance of the specific temperatures and pressures.

Unlike *Schulze*, Appellants’ specification at p. 5, ll. 15-24 and page 11, ll. 15-20, teaches wizards provided by the invention enable reuse, extensibility, and information sharing among host-wizards and sub-wizards. Further, Appellants have not asked the Office to ignore Appellants’ specification in the reasons provided to the Office. Rather, Appellants have requested the Office to provide a reasonable interpretation of wizard as in the context of the

pending claims and Appellants' specification. The Office comment that "applicant is free to copy text from the specification into the claims if applicant desires specific limitation on how 'wizard' is to be interpreted," is noted. Appellants only request that "wizard," as utilized in Appellants' specification and the pending claims, is interpreted reasonably as required by the Manual of Patent Examining Procedure and the Court. MPEP § 2111.01.

The Office's conclusion that a host wizard is an Active Server Page (seismic.asp) is unreasonable. *See, e.g.*, Final Office Action mailed 12/05/06 at p. 3. The Office's interpretation of host-wizard is beyond the ordinary and customary meaning of the term as utilized in claims 7, 10, and 11 and Appellants' specification. As taught by Fedorov, at page 423, ll. 30-37, the Active Server Page (seismic.asp) provides server-side calculations, checks a uniform resource locator (URL) for data to process, and loads a "wizard dialog box." A reasonable reading of Fedorov does not support the Office's interpretation that seismic.asp is a host-wizard. Seismic.asp is not a wizard that defines an extension interface to respond to navigation events, where the extension interface enables recursive navigation between the host-wizard and a web component. Seismic.asp is not a multistep process that is controlled by a user's navigation of screens. Rather, seismic.asp merely loads a wizard dialog box that collects data, which is processed by a server executing seismic.asp.

The Office's conclusion that a user interface that integrates a web component into a host wizard by utilizing the extension interface to perform recursive navigation between the web component and host wizard as recited in independent claims 7, 10, and 11 is anticipated by an html page (equakeget.htm) as in Fedorov is unreasonable. *See, e.g.*, Final Office Action mailed 12/05/06 at p. 3. Fedorov, at page 423, ll. 25-32, teaches the user interface for the

wizard dialog box loaded by seismic.asp is provided by equakeget.htm. Equakeget.htm generates the user interface for the wizard dialog box. Equakeget.htm does not integrate a web component into the host wizard by utilizing an extension interface provided by the host wizard to perform recursive navigation between the web component and the host wizard. At best, equakeget.htm only provides a user interface based on navigation within the wizard dialog box. Nothing in Fedorov teaches a user interface based on navigation between a host wizard and a web component provided by an extension interface of the host wizard.

Additionally, the Office reasons that the “Back” and “Next” of the wizard dialog box controls in Fedorov provide recursive navigation between a host wizard and web component. Contrary to the Office’s allegation, Fedorov, at page 424, expressly indicates that the “Back” and “Next” controls are used to navigate between panes of the dialog box wizard. Nothing in Fedorov, teaches that the “Back” and “Next” controls provide recursive navigation between a host wizard and a web component.

Unlike Fedorov, the invention of independent claims 7, 10, and 11, require, among other things, a host-wizard that defines an extension interface that responds to navigation events to provide recursive navigation between a web component and the host wizard. Fedorov fails to teach a host wizard and integration of the host wizard and a web component in a manner that provides recursive navigation between the host wizard and the web component when generating the user interface. Accordingly, for at least the reasons set forth above, Appellant respectfully requests withdrawal of the anticipation rejection and allowance of independent claims 7, 10, and 11.

B) Rejection under 35 U.S.C. § 102(e) over Gauthier *et al* (U.S. Patent No. 6,574,791).

Claims 1-3, 8, 9, and 14 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Gauthier. Appellant respectfully traverses this rejection, as hereinafter set forth.

(i) Claims 1, 8, and 9

With respect to independent claims 1, 8, and 9, Gauthier fails to teach, among other things, “invoking said one or more sub-wizard components during said host-wizard component execution; and transferring control from said host-wizard to said one or more sub-wizard components.”

The Office indicates that Gauthier, at col. 9, l. 56-col. 10, l. 5, anticipates the claimed requirements. The section cited by the office action discloses:

“The WizardManager class defines objects which control the execution of multiple subwizards within the target wizards. This class also references the GUI elements making up the overall target wizard. This class preferably includes constructor methods for creating a new WizardManagerFrame object, a new WizardManagerButtonPanel object, a new WizardManagerSelectionPanel object, and a new WizardManagerLogoPanel object, and constructor methods that creating these objects from specified existing meta data. These constructor methods would call corresponding constructor methods on the corresponding class. This class also preferably includes an AddWizard() method for adding new subwizards to the overall wizard skeleton. This class also preferably includes methods for launching the overall wizard, and launching selected subwizards. The class also preferably includes a describe content method used to export meta data from the class objects.”

In Gauthier, a “WizardManager” controls the execution of the subwizards which define the Target wizard, at col. 13, ll. 25-35. The execution of the subwizards is centralized and coordinated by the WizardManager. The Gauthier disclosure differs from the invention defined by claims 1, 8, and 9 because the control functions of Gauthier are centralized and not distributed as required by claim 1, 8, and 9. The Office further contends that Gauthier, at col. 14, l. 9 through col. 15, l. 24, teaches the claimed “transferring of control to the subwizard.”

Gauthier discloses WizardState objects that are utilized to generate code that implement the wizard functions. Nothing in the cited section discloses control is passed to the sub-wizard during the execution of the host wizard. The Office contends that control must be passed to the sub-wizard. However, a reasonable reading of Gauthier, at col. 9, ll. 55-58, would lead one of ordinary skill in the art to conclude that control is maintained by the Wizard Manager Class.

Unlike Gauthier, independent claims 1, 8, and 9 require, among other things, a host-wizard to transfer control directly to a sub-wizard, and the sub-wizard to control its own execution. Gauthier fails to teach a transfer of control to one or more sub-wizard components. Accordingly, for at least the reasons set forth above, Appellants respectfully request withdrawal of the anticipation independent claims 1, 8, and 9 is allowable over Gauthier, and the anticipation rejection of claim 1 should be withdrawn.

(ii) Claim 2

Claim 2 depends from claim 1 and further defines novel features of the claimed invention. Accordingly, claim 2 is allowable by virtue of its dependence on claim 1. Additionally, claim 2 is allowable because Gauthier fails to teach “wherein one or more sub-wizard components are browser based object components.” The Final Office Action has referenced Gauthier, at col. 18, l. 66-col. 19, l. 30, to anticipate the claimed requirement. Final Office Action mailed 12/5/06. The section cited by the Office discloses:

“Turning now to FIG. 6, the preferred embodiment implementation of the WizardMetaDataManager 128 is illustrated in more detail. As discussed above, the WizardMetaDataManager 128 is used to persist and retrieve target wizard meta data. In the preferred embodiment, the WizardMetaDataManager 128 implementation includes a WizardMetaDataManager object, a WizardMetaData object, a WizardMetaDataFormDescription object, a WizardMetaDataPanel object, and a WizardMetaDataFrame object.

The WizardMetaDataManager object preferably includes a ReuseExistingMetaData() method and a SaveWizardMetaData(). The SaveWizardMetaData() displays the WizardMetaDataPanel object which prompts the developer to select the meta data to be stored and specify a wizard meta data file in a form specified by the WizardMetaFormDescription object. The ReuseExistingMetaData() method displays the WizardMetaDataPanel and prompts the user to select a set of wizard meta data. The method retrieves the wizard meta data and walks through it to recreate an **internal** set of wizard framework objects.

The WizardMetaData object includes the actual meta data for the target wizard. This would preferably include all of the components of the target wizard and their interrelationships. The storage would preferably be done in a **language neutral format** to facilitate ease of restoring. The WizardMetaFormDescription object describes the format used for storing the wizard meta data. For example, the wizard meta[1] data could be described using extensible markup language (XML) and an accompanying wizard framework specific document content description. The WizardMetaDataPanel object and a WizardMetaDataFrame object provide the GUI interface to the developer."

Gauthier teaches a WizardMetaDataManager that is able to retrieve stored meta data, the meta data preferably being stored in a language neutral format. Gauthier further discloses that it is possible to describe the meta data using XML and a specific document content description. XML is an example of the language neutral format that aids in defining the meta data. Gauthier, at col. 14, ll. 9-12, further discloses that each sub-wizard includes a Wizard object, a WizardState object, a WizardDefault object, a WizardStateController object, WizardPanel object(s), and a WizardCodeGenerator object. Gauthier discloses that the XML format is utilized to represent metadata associated with a target wizard. Gauthier does not teach that XML is utilized to define a target wizard. As described in Gauthier XML is a storage format and not a browser based object component that receives control from a host wizard during the execution of the host wizard. Accordingly, Gauthier fails to disclose sub-wizards that include browser based object components. Appellants' specification, at page 4, lines 1-10 and 21-24, page 5, lines 20-24, and page 11, lines 20-23 teaches that browser based object components include HTML or web pages that provide web wizards.

Unlike Gauthier, dependent claim 2 requires, among other things, sub-wizard components that are browser based object components. Gauthier fails to teach browser based object components. Therefore, for at least the above reasons, Appellants respectfully request withdrawal of the anticipation rejection and allowance of claim 2.

(iii) Claim 3

Claim 3 depends from claim 1 and further defines novel features of the claimed invention. Accordingly, claim 3 is allowable by virtue of its dependence on claim 1. Additionally, claim 3 is allowable because Gauthier fails to disclose “wherein one or more sub-wizard components are operating system based application component object extensions.” The Office has referenced Gauthier, at col. 6, ll. 42-46, as anticipating the claimed requirement. The section cited by the office action discloses:

“The operating system 122 provides the basic functionality that controls the computer system 100. Operating system 122 can comprise any suitable operating system, such as IBM’s AS/400, OS/2, Microsoft’s Windows, Java and the various flavors of UNIX”

In this section, Gauthier teaches the use of an operating system and the different types of operating systems. Gauthier, at col. 7, ll. 45-58, further discloses a wizard framework that defines the basic elements of a wizard. This framework defines the core functions of the solution, those elements that are required and cannot be extended by a developer. The framework also defines extensible functions of the solution, those that can be customized and extended by the developer. The customization/extension quality of framework mechanisms is extremely valuable because the cost of customizing or extending a framework is much less than the cost of replacing or reworking an existing solution.

Although Gauthier refers to extensions of particular functions, these types of extensions are limited specifically to objects within the wizard framework. There is no teaching of an extension that utilizes operating system based application components. Appellants' specification, at pp. 14-15, illustrates that an operating system based extension comprises a traditional object wizard component extension and is created in an operating environment that differs from the browser based object component which comprises the HTML component wizard extension.

Unlike Gauthier, dependent claim 3 requires, among other things, sub-wizard components that are operating system based application component object extensions. Gauthier fails to teach operating system based application component object extensions. Therefore, for at least the above reasons, Appellants respectfully request withdrawal of the anticipation rejection and allowance of claim 3.

(iv) Claim 14

With respect to independent claim 14, Gauthier fails to teach, among other things, "at least one navigation component on each of said first and second wizards, said navigation components allowing sequential progression or regression through said first and second wizards to chain said second wizard to said first wizard to guide a user through the first and second tasks."

The Final Office Action has referenced Gauthier, at col. 9, l. 55–col. 10, l. 5, col. 10, ll. 57-67, and col. 14, ll. 9-14. Gauthier, at col. 10, ll. 20-30, teaches the WizardManagerButtonPanel class provides a panel with a plurality of buttons for use on the target wizard interface. These buttons would typically include standard GUI interface buttons, such as BACK, NEXT, FINISH, CANCEL and HELP. However, the buttons are not on **each wizard**. Instead, the buttons are centralized using a WizardManagerFrame, col. 10, ll. 5-40. Furthermore, as noted above the

execution and control of the subwizards of Gauthier are centralized; so, the subwizards are not chained through the use of navigational components. Gauthier, at col. 2, ll. 4-13 and col. 11, ll. 54-63, teaches that a wizard guides a user through a task and provides the ability to present GUI components that collect information from the user. However, Gauthier does not teach how to integrate two separate and distinct wizards in the manner claimed in independent claim 14.

Unlike Gauthier, independent claim 14 requires, among other things, at least one navigation component on each of said first and second wizards, said navigation components allowing sequential progression or regression through said first and second wizards to chain said second wizard to said first wizard to guide a user through the first and second tasks. Gauthier fails to teach navigation components on each wizard that allow sequential progression or regression through said first and second wizards to chain said second wizard to said first wizard to guide a user through the first and second tasks. Therefore, for at least the above reasons, Appellants respectfully request withdrawal of the anticipation rejection and allowance of independent claim 14.

C) Rejection under 35 U.S.C. § 103(a) over Fedorov (Professional Active Server Pages 2.0,” 1998, Wrox Press Ltd) and Gauthier *et al* (U.S. Patent No. 6,574,791).

Title 35 U.S.C. § 103(a) declares, a patent shall not issue when “the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.” The Supreme Court in *Graham v. John Deere* counseled that an obviousness determination is made by identifying: the scope and content of the prior art; the level of ordinary skill in the prior art; the differences between the claimed invention and prior art references; and secondary considerations. *Graham v. John Deere Co.*, 383 U.S. 1 (1966). To support a finding of obviousness, the initial burden

is on the Office to apply the framework outlined in *Graham* and to provide some reason, or suggestions or motivations found either in the prior art references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the prior art reference or to combine prior art reference teachings to produce the claimed invention. See, *Application of Bergel*, 292 F. 2d 955, 956-957 (CCPA 1961). Recently, the Supreme Court elaborated, at pages 13-14 of the *KSR* opinion, it will be necessary for [the Office] to look at interrelated teachings of multiple [prior art references]; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by [one of] ordinary skill in the art, all in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the [patent application]." *KSR v. Teleflex*, No. 04-1350, 550 U.S. ____ (2007).

(i) Claims 16 and 18-19

Claims 16 and 18-19 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,574,791 to Gauthier in view of Fedorov. Appellants respectfully traverse the rejection, as hereinafter set forth.

Dependent claims 16 and 18-19 further define novel features of the claimed embodiments and each depend either directly or indirectly, from one of independent claims 1, 8, and 9. Accordingly, for at least the reason set forth above with respect to independent claims 1, 8, and 9, dependent claims 16 and 18-19 are believed to be in condition for allowance by virtue of their dependency. 37 C.F.R. 1.75(c). As such, withdrawal of the 35 U.S.C. § 103(a) rejection of dependent claims 16 and 18-19 is respectfully requested.

Furthermore, with respect to claims 16 and 18-19, Gauthier and Fedorov, singularly or in combination fail to teach or suggest, among other things, “passing a property bag between said host wizard and said one or more sub-wizard components.”

The Office concedes that Gauthier fails to disclose or suggest the claimed passing of a property bag. However, the Office contends that Fedorov remedies Gauthier’s deficiency and teaches or suggests the claimed passing of a property bag. Appellants respectfully disagree. Fedorov, at page 423, expressly teaches that seismic.asp provides server-side calculations. Seismic.asp is not a host wizard. The Office contends that seismic.asp anticipates the claimed host wizard. Fedorov, expressly discloses that Seismic.asp redirects the browser to the server when calculations are performed. Furthermore, Gauthier maintains centralized execution of the wizards so information does not need to be passed in manner described in claims 16 and 18-19.

Unlike Fedorov and Gauthier, singularly or in combination, the claimed embodiment requires passing a property bag between a host-wizard and other wizards when navigating and passing control among the wizards. Fedorov teaches utilizing a server to perform calculations on information collected by a wizard. There is nothing in Fedorov or Gauthier, singularly or in combination, that teaches or suggests passing a property bag between at least two wizards when navigating and transferring control between the wizards. Accordingly, for at least the foregoing reasons, Appellants respectfully request the withdrawal of the obviousness rejection and allowance of claims 16 and 18-19.

Appellants respectfully submit that claims 1-3, 7-11, 14, 16, and 18-19 are in condition for allowance. As such, Appellants respectfully request that the rejection of the claims be reversed and that a timely Notice of Allowance be issued in this case. Should there be any unresolved matters, please contact the undersigned.

Respectfully submitted,

Date: 03 August 2007.

/MONPLAISIR HAMILTON

Monplaisir Hamilton
Reg. No. 54,851

SHOOK, HARDY, & BACON L.L.P.
2555 Grand Blvd.
Kansas City, MO 64108-2613
Tel.: 816/474-6550
Fax: 816/421-5547

Attorney Docket No. MFCP.88142

VIII. CLAIMS APPENDIX

1. (Previously Presented) A method for use in a computing environment for extending a wizard comprising:

generating a host-wizard component;

generating one or more sub-wizard components;

invoking said one or more sub-wizard components during said host-wizard component execution; and

transferring control from said host-wizard to said one or more sub-wizard components.

2. (Previously Presented) A method as recited in claim 1 wherein said one or more sub-wizard components are browser based object components.

3. (Previously Presented) A method as recited in claim 1 wherein said one or more sub-wizard components are operating system based application component object extensions.

Claims 4-6. (Canceled).

7. (Previously Presented) A method for use in a computing environment for extending a wizard comprising:

generating a host wizard that defines an extension interface to respond to navigation events;

generating a web component comprising:

a web page, said web page containing a header area, a wizard control area and a control interface area;

one or more object module functions, said object module functions enabling navigation; and

said control interface area having navigation control to recursively navigate within said web component and to said host wizard, by utilizing said one or more object module functions;

generating a user interface that integrates said web component into said host wizard by utilizing the extension interface to perform recursive navigation between said web component and said host wizard; and

utilizing an information container to exchange informational items between said web component and said host wizard.

8. (Previously Presented) A machine readable medium having machine executable instructions for performing a method for extending a wizard comprising:

generating a host-wizard component;

generating one or more sub-wizard components;

invoking said one or more sub-wizard components during said host-wizard component execution; and

transferring control from said host-wizard to said one or more sub-wizard components.

9. (Previously Presented) A computer system having a processor, a memory and an operating environment, the computer system operable to execute a method for use in a computing environment for extending a wizard comprising:

generating a host-wizard component;

generating one or more sub-wizard components;

invoking said one or more sub-wizard components during said host-wizard component execution; and

transferring control from said host-wizard to said one or more sub-wizard components.

10. (Previously Presented) A machine readable medium having machine executable instructions for performing a method for extending a wizard comprising:

generating a host wizard that defines an extension interface to respond to navigation events;

generating a web component comprising:

a web page, said web page containing a header area, a wizard control area and a control interface area;

one or more object module functions, said object module functions enabling navigation; and

said control interface area having navigation control to recursively navigate within said web component and to said host wizard, by utilizing said one or more object module functions;

generating a user interface that integrates said web component into said host wizard by utilizing the extension interface to perform recursive navigation between said web component and said host wizard; and

utilizing an information container to exchange informational items between said web component and said host wizard.

11. (Previously Presented) A computer system having a processor, a memory and an operating environment, the computer system comprising:

a host wizard that defines an extension interface to respond to navigation events;

a web component comprising:

a web page, said web page containing a header area, a wizard control area and a control interface area;

one or more object module functions, said object module functions enabling navigation; and

said control interface area having navigation control to recursively navigate within said web component and to said host wizard, by utilizing said one or more object module functions;

a user interface that integrates said web component into said host wizard by utilizing the extension interface to perform recursive navigation between said web component and said host wizard; and

an information container to exchange informational items between said web component and said host wizard.

Claims 12-13. (Canceled).

14. (Previously Presented) A computer system having a processor, a memory and an operating environment, the computer system comprising:

a first wizard having panels to guide a user through a first task;

a second wizard having panels to guide a user through a second task; and

at least one navigation component on each of said first and second wizards, said navigation components allowing sequential progression or regression through said first and second wizards to chain said second wizard to said first wizard to guide a user through the first and second tasks.

Claim 15. (Canceled)

16 (Previously Presented) A method as recited in claim 1, wherein transferring control from said host-wizard to one or more sub-wizard components, further comprises:

passing a property bag between said host-wizard component and said one or more sub-wizard components.

Claim 17 (Canceled).

18 (Previously Presented) A machine readable medium as recited in claim 8, wherein transferring control from said host-wizard to one or more sub-wizard components, further comprises:

passing a property bag between said host-wizard component and said one or more sub-wizard components.

19 (Previously Presented) A computer system as recited in claim 9, wherein transferring control from said host-wizard to one or more sub-wizard components, further comprises:

passing a property bag between said host-wizard component and said one or more sub-wizard components.

Claim 20 (Canceled).

IX. EVIDENCE APPENDIX

Not applicable

X. RELATED-PROCEEDINGS APPENDIX

Not applicable